

# A comparison of ABR and UBR to support TCP traffic

Sam Manthorpe and Jean-Yves Le Boudec

**Abstract**— This paper compares the performance of ABR and UBR for providing high-speed network interconnection services for TCP traffic. We test the hypothesis that UBR with adequate buffering in the ATM switches results in better overall goodput for TCP traffic than explicit rate ABR for LAN interconnection. This is shown to be true in a wide selection of scenarios. Four phenomena that may lead to bad ABR performance are identified and we test whether each of these has a significant impact on TCP goodput. This reveals that the extra delay incurred in the ABR end-systems and the overhead of RM cells account for the difference in performance. We test whether it is better to use ABR to push congestion to the end-systems in a parking-lot scenario or whether we can allow congestion to occur in the network. Finally, we test whether the presence of a ‘multiplexing loop’ causes performance degradation for ABR and UBR. We find our original hypothesis to be true in all cases. We observe, however, that ABR is able to improve performance when the buffering inside the ABR part of the network is small compared to that available at the ABR end-systems. We also see that ABR allows the network to control fairness between end-systems.

## I. INTRODUCTION

In this paper we examine the suitability of two different classes of Asynchronous Transfer Mode (ATM) virtual connection (VC) for providing a network interconnection service. The Available Bitrate (ABR) and Unspecified Bitrate (UBR) services have been designed with such an application in mind, since they provide a link layer service for traffic that does not require a delay guarantee. The majority of inter-LAN traffic on today's networks is created using the Transport Control Protocol (TCP) [1] at layer 4. In recent measurements on an EPFL network [2], TCP accounted for 93% of all inter-LAN traffic. We therefore study the performance of TCP traffic over ABR and UBR, using the high-speed network simulation tool, STCP.

The ABR service has been introduced by the ATM Forum to cater for traffic sources that demand low loss ratios but can accept large delays. The idea behind ABR is to exploit the bandwidth that is unused by other CBR and VBR services, as illustrated in figure 1. It is known that video traffic, for example, has long range dependence that makes it difficult to obtain statistical multiplexing gain by multiplexing several connections on a link [3], [4]. ABR, on the other hand, can take advantage of the bandwidth unused by a video connection, giving it to the data traffic sources.

ABR is a congestion avoidance mechanism. Feedback is provided to the User Network Interface (UNI) that indicates at what rate a source is allowed to transmit. The ATM

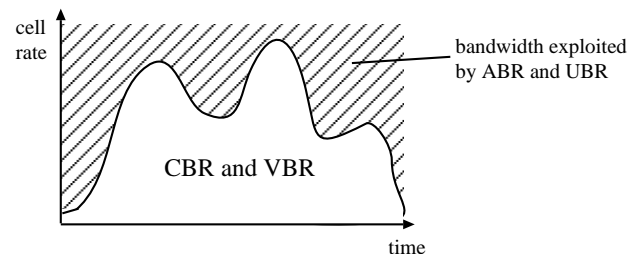


Fig. 1. ABR exploits bandwidth unused by other traffic classes

switches in the network send this feedback by modifying fields in RM cells that are sent periodically by the sources. There are two modes possible: *binary mode* and *explicit rate mode*. In binary mode, a switch indicates whether it is experiencing (or about to experience) congestion or not. The ATM Forum has defined a fairly complex set of rules to define how a source should behave upon reception of this feedback information [5]. This mode of operation is intended primarily for legacy switches. Analytical studies of binary mode ABR can be found in [6]. With explicit rate mode, the switches signal explicitly at what rate a source is allowed to send; this rate is usually calculated so as to obtain optimal utilisation of network resources and fairness between sources. Explicit rate mode involves more complexity in the switches, but offers better performance and avoids the oscillatory behaviour that binary mode is prone to. Both schemes require an algorithm running in the switch to decide what feedback should be sent to the source. In the case of binary mode, this is usually a simple mechanism based upon the buffer occupancy whereby “no-increase” signals are sent if the buffer occupancy is above a certain threshold and “decrease” signals are sent if it is above another threshold. In the case of explicit rate mode, the mechanism can be more complex, since an explicit rate has to be calculated. Several mechanisms have been proposed for this (see [7], [8], [9]). In this contribution, we consider explicit rate mode. We adopt the ERICA algorithm [8] in the ATM switches to control the feedback messages (see section I-A for more details). ABR required priority queueing in the switches to separate the ABR traffic from other flows of delay-sensitive traffic classes.

The Unspecified Bitrate (UBR) class is a simpler service that requires no traffic parameters and offers no congestion avoidance. All traffic sent by an end-system is accepted by the network, but no guarantee is given that it will be successfully transported to its destination, thus allowing the network to discard it in case of congestion. UBR is identical to the “best-effort” concept. Priority buffering is required

in the switches to implement the service and some kind of congestion avoidance mechanism required at a higher protocol layer (which is normally the case).

The UBR service may be enhanced using the Early Packet Discard (EPD) mechanism [10] to reduce packet loss in congested buffers and to reduce the amount of resource waste from cells whose parent packets have been lost [11], [12]. EPD works by only accepting cells into a buffer if either the buffer occupancy is less than a certain threshold (which we will call the EPD threshold), or if other cells belonging to the same packet have already been stored in the buffer. Such a service is often called UBR+.

One may expect that since ABR avoids loss in the ATM network that the user whose traffic goes over an ABR based LAN interconnection VC will experience better performance than those whose traffic goes over a UBR connection. However, this is not necessarily the case since there are several factors that may lead to degradation of TCP performance over ABR. Apart from the traffic overhead of RM cells, ABR introduces an extra queueing stage at the edge of the ATM network, which can increase end-to-end delay. The ABR control loop is designed to slow down sources in case of congestion; however for LAN interconnection, the source that ATM controls is simply a router and ABR is unable to signal to the *real* traffic sources (workstations, PCs etc.) the rate at which they should transmit. This means that congestion may be avoided in the ATM part of the network, but not for the end-to-end TCP traffic flows. Furthermore, the introduction of the ABR congestion control loop *inside* the TCP congestion control loop may lead to undesirable interactions between the two that may degrade performance. Finally, for ABR to properly achieve its objectives of high link utilisation and low loss, many parameters must be correctly tuned. So it is not certain that ABR is the correct VC class for interconnection services. In this paper we perform extensive simulation studies which show that in most cases, UBR results in better overall TCP performance. We also consider the Early Packet Discard (EPD) scheme for improving the performance of the UBR service.

### A. Simulator

For these studies, the STCP 3.0 simulator, developed as part of project 317 with the Swiss PTTs, was used<sup>1</sup>. STCP is a detailed simulator of TCP over ATM. It was developed [13] using the original source code of Berkeley 4.4 UNIX TCP (see [14] for a complete description), which incorporates most of the features found in today's TCP implementations, most importantly slow-start, congestion detection, congestion avoidance and fast-retransmit [15]. Using the real source code ensured a high degree of accuracy in the model. Moreover, the custom implementation in C means that STCP is very fast compared to most commercially available simulation packages. The simulator allows work-

<sup>1</sup>STCP is freely available on the WWW at <http://lrcwww.epfl.ch/~manthorp/stcp/>

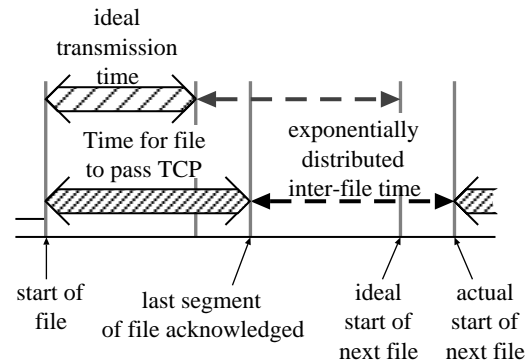


Fig. 3. Influence of the TCP protocol on the on-off traffic source

stations, background sources (Markov chains, Poissonian, On-Off and constant bitrate) queues and priority queues to be linked together to form different simulation networks.

In these studies, traffic was generated by workstation models, using a TCP/IP/AAL5/ATM protocol stack. Figure 2 shows the architecture of a pair of model workstations. In STCP, workstations always come in pairs and a distinction is made between source and destination.

A source initiates connections to the destination and sends data on these connections. A destination waits for connection requests and once a connection has been established, acts as a sink for the data that the source sends, sending ACK packets according to the TCP protocol. Traffic is fed to the TCP layer by an on-off traffic source. When this source goes into the on state, it opens a socket to connect to the destination. This causes TCP to request a connection with the destination by sending a SYN packet. After the normal handshaking sequence, the connection is established and the traffic source gives a random, geometrically distributed amount of data to the TCP socket to send to the destination. Once TCP has sent all of this data, the connection is released, again using the normal handshaking sequence, and the on-off source model goes into the off state. It then waits in the off state for an exponentially distributed amount of time, after which it goes back into the on state and the procedure is repeated. The on-off source model is almost the same as the classical on-off models seen in the literature, *except* that it cannot make the transition from the on state to the off state until TCP has sent all of the data. The time it takes to send this data depends on the network and the speed of the destination. This is illustrated in figure 3.

In these studies, the network performance was judged based primarily on the TCP goodput.

### B. Overview

This paper aims at verifying the hypothesis that UBR with adequate buffering in the ATM switches results in better overall goodput for TCP traffic than explicit rate ABR for LAN interconnection. To test this, we perform extensive simulation studies of different network interconnection sce-

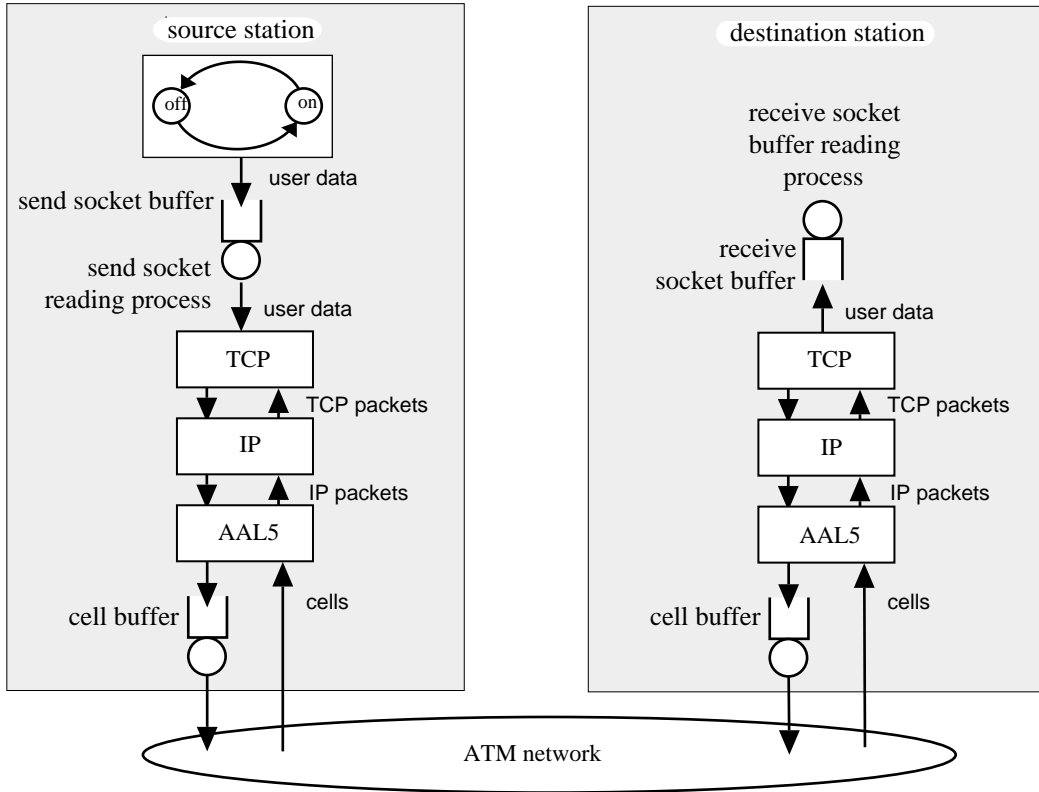


Fig. 2. Workstation pair

narios, using ABR, UBR and UBR with EPD for the interconnection VCs. We start in the following section by introducing a simple, one-hop LAN interconnection model that is adopted in the first half of this paper. Section III considers the dimensioning of the ABR service so as to give good LAN interconnection performance. Then, in section IV we examine the three services' behaviour under varying non-ABR traffic load in the network. We see that UBR and UBR with EPD resulted in better TCP performance than ABR, for both long and short distance VCs and medium and large packets. In section V we identify four phenomena that may contribute to the degradation of TCP performance over an ABR VC and test each of these to see if they do indeed account for the reduction in goodput. Results show that the RM cell overhead and the extra queueing delay incurred in the ABR end systems account for this performance degradation. Section VI then further tests our hypothesis by examining the influence of different network parameters on the performance curves for ABR, UBR and UBR with EPD in the one-hop network. It is shown that UBR outperformed ABR from the point of view of TCP goodput for different background traffic parameters and numbers of TCP sources. EPD offered even better performance. We found, however, that for small switch buffers, ABR was able to improve TCP performance, reducing loss in the network by exploiting the larger buffers at the ABR end-systems.

Section VII considers two, more complex scenarios involving more than 1 hop: a five-node parking-lot configuration and a triangular network. Again, it is shown how UBR gave

better performance than ABR. Finally, there is a discussion of the results.

## II. INTERCONNECTION MODEL

In this section we present the generic LAN interconnection scenario that is simulated in the following sections. To start with we consider a network with only one hop, so as to have a clear picture of the mechanics of the network. In later sections we consider more complex scenarios.

We adopt the heterogeneous network shown in figure 4. Two shared-medium LANs are interconnected by an ATM network with just one switch. Only one switch was modelled since the aim was to have a clear view of the interaction between the ABR control units and the ERICA algorithms in the switch. More complex models are studied later.

Interconnection between the two LANs is provided by a VC connecting the two control units (a control unit is analogous to a router or an Interworking Unit (IWU)). The VC may be of the ABR or UBR class. The ingress buffer is where cells waiting to be transmitted on an ABR connection are stored. The service rate of this buffer is determined by the ABR protocol and so significant queues may develop when the available bitrate is low. Each ABR control unit comprises an ingress queue and an egress queue. To avoid confusion, the reader should note that in these simulations the ingress queue is equivalent to the ATM Forum's Source End System (SES) and the egress queue is equivalent to a Destination End System (DES). For UBR services, no

queueing occurs in the control units, since all traffic is sent as soon as it arrives.

$d_{net}$  is the propagation delay between a node in the ATM network, which means between the two ABR control units and the connected switch. The total end-to-end delay,  $e$ , is:

$$e = 2 \cdot d_{net} + d_{switch} + 2 \cdot d_{LAN} \quad (1)$$

where  $d_{switch}$  and  $d_{LAN}$  are the internal delay in the switch and the delay in the LAN and are equal to  $450 \mu s$  and  $10 \mu s$  respectively. In this paper we consider three interconnection scenarios for the ABR part of the network: LAN ( $d_{net} = 10 \mu s$ ), MAN ( $d_{net} = 5 ms$ ) and WAN ( $d_{net} = 50 ms$ ).

We do not specify the technology of the LANs, which are modelled as a single queue for all traffic going through them, both to and from the control unit.

Data traffic is generated by TCP sources connected to LAN 1, which is sent to TCP destinations on LAN 2. Traffic in the reverse direction consists only of ACK and connection set-up and release handshaking packets.

Our model has the following important features:

- We model the traffic at the connection level instead of assuming infinite sources. This means that we model the effect of a new connection starting up or being released on the rest of the traffic.
- The ABR loop does not go as far as the TCP source itself, but to a router, contrary to previous TCP over ABR studies.
- Very accurate TCP simulation using STCP.

The speed of STCP allowed us to run simulations for hundreds of seconds of model time, typically two orders of magnitude greater than previous studies.

#### A. Switch

Figure 5 shows the architecture of the switches used in these simulations. The switch is output buffered, with priority queueing used to ensure that the ABR or UBR traffic does not cause congestion for the non-ABR/non-UBR traffic (for example VBR video traffic). We call the non-ABR/non-UBR traffic the background traffic and it is detailed below. The capacity of the background traffic queue is 128 cells and the capacity of the ABR/UBR queue is 8192 cells. This is typical of ATM LAN switches today; the Fore ASX-200, for example has a per-port buffer capacity of 8928 cells. Later we will investigate the role of the buffer capacity parameter on the relative performance of ABR and UBR. Note that from the switch's point of view, the only difference between ABR and UBR is that ABR requires on-line traffic measurement and processing of RM cells. The queueing mechanisms themselves are exactly the same for ABR and UBR. In the case of UBR with EPD, the EPD mechanism is also implemented in the UBR queue.

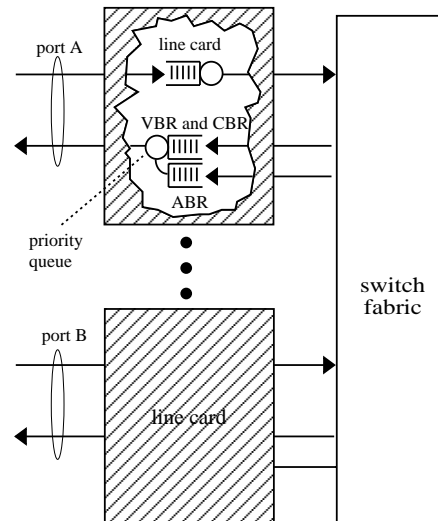


Fig. 5. Switch model

#### B. ERICA

STCP uses the ERICA and ERICA+ algorithms for calculating the explicit rate feedback in the switches for explicit-rate mode ABR. The algorithm aims at maintaining an equilibrium in the switch in which the utilisation of the available bandwidth is close to 1 and the cell loss is small. In these simulations, the algorithm was found to achieve these objectives. The backwards RM cells are modified rather than the forward ones so as to decrease the time between the switch sending feedback to the source and the source receiving it. ERICA modifies the ER fields of the RM cells in the backwards direction only, to reduce the feedback time to the source. In addition to the basic ERICA algorithm, the scheme includes many innovations for improving performance, which are described in [8]:

The performance of the ABR control loop depends quite heavily on the choice of parameters used. After experimentation, the parameters shown in table I were found to give good performance. The parameter that has the strongest influence on performance is the rate measurement window size,  $\omega_{ERICA}$ . Since rate allocations are calculated once per window, the window size determines not only the accuracy of the measurements but the delay of the feedback loop. If the window size is too small then the estimation of the arrival rate and of the number of active VCs can be inaccurate, but if it is too large the risk of bifurcation increases.

#### C. Background traffic

In order to observe the switch's ability to flow control the ABR traffic in the presence of non-ABR traffic, a background load was fed into switch 1. The non-ABR traffic has priority over the ABR traffic and the ERICA algorithm, based upon on-line measurements, should control the ABR sources so that if the resources are being used by the non-ABR traffic, they adjust their rates so as to avoid loss in the ABR buffer in the switch. The background traffic itself con-

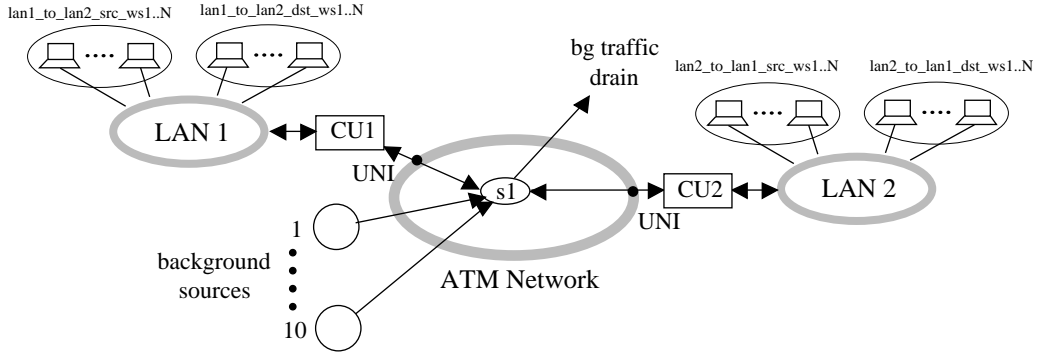


Fig. 4. Network model

Parameter	value
Measurement window size, $\omega_{ERICA}$	2 ms
Target utilisation	0.95
$\alpha$ for load averaging	0.0625
$\delta$ decay factor for VC contribution averaging	0.25
ERICA + queue control options	disabled

TABLE I

DEFAULT ERICA PARAMETERS USED IN THESE STUDIES

sisted of 10 classic on-off sources; that is two state Markov modulated sources, where cells are created at a constant rate,  $p$ , (also known as the peak rate) in one state and no cells are created in the other. The time in each state was exponentially distributed. The peak rate of each source,  $p_{bg}$  was set to  $1.25 \cdot l/N_{bg}$ , where  $l$  is the link rate and  $N_{bg}$  is the number of sources. The mean burst length,  $b_{bg}$  was by default 200 cells. The mean rate of a single source is denoted by  $\mu_{bg}$ . The mean rate of the ensemble of sources is expressed in terms of the *background utilisation*,  $\rho_{bg}$ , which is equal to  $N_{bg} \cdot \mu_{bg}/l$ .

#### D. LAN

LAN 1 contains  $N$  workstations that initiate TCP connections to LAN 2. LAN 2 contains  $N$  workstations that wait for connection requests from LAN 1 and act as sources for the data traffic. The default value for  $N$  in this study was 10. Each workstation that initiates TCP connections tries to do so at a rate of 2 calls per second and the mean file length for the connection was 453125 bytes. Note that we say that the workstation “tries” to transmit at a certain rate, since, as described above, the rate at which it *actually* creates traffic depends on the network conditions. In the following we will call the traffic that a workstation tries to create the *offered load*. When a file is being transmitted, the data is loaded into the output TCP socket buffer, which has a capacity of 61444 bytes, at a rate of 50 Mbits/s. At the destination, data is read from the socket buffer at a rate

of 100 Mbits/s, so that window advertising does not have an influence on the traffic. The peak cell rate of the NIC is 100 Mbits/s. The above parameters resulted in an external traffic with an offered load of 72 Mbits/s and which, by today's standards, corresponds to a large LAN<sup>2</sup>.

### III. ABR SERVICE DIMENSIONING

Before we compare the performance of an ABR based interconnection service with other approaches, we first look at dimensioning the service in order to obtain good performance. In order to do this, two parts of the network have to be properly dimensioned: the ingress node (where the UNI is) and the switches. For explicit rate mode of operation, the ingress node has the following parameters that should be considered:

- Peak Cell Rate (PCR)
- Minimum Cell Rate (MCR)
- Ingress buffer capacity

In order to be able to exploit as much of the available ATM resources as possible, we set the PCR to the link rate (i.e. 145 Mbits/s). The MCR should be as low as possible to avoid unnecessary reservation of bandwidth in the network. However, it is advantageous to have a certain amount of bandwidth reserved, since this avoids having to send RM cells out-of-rate and improves the response time of a previously idle ABR connection when new data arrives and needs to be sent. The lower this bandwidth, the slower the ABR control loop's response time when more bandwidth is needed, since it has to wait until it is next allowed to send a cell. Since TCP traffic normally adopts the ‘slow-start’ algorithm, high bandwidth is not necessary during the first few RTTs of a connection, when it is ‘warming-up’. In this scenario we set the MCR to 1 Mbits/s which gives a fairly good response time to newly arriving data.

The ingress queue is where traffic to be sent on the ABR connection is queued when the connection's ACR is less than the arrival rate. Dimensioning of the ingress queue is an important issue for providing good ABR performance.

<sup>2</sup>With the increasing use of multimedia applications and the increasing amount of long-distance traffic, such figures may well correspond to a medium size LAN in the future).

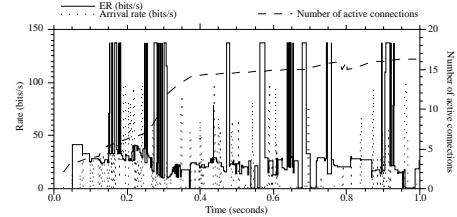
Before explaining this dimensioning work, it is first important to remember that ABR is not a mechanism for reducing end-to-end loss. At a network level, what ABR actually does is not to avoid congestion but to relocate it. ABR can be very effective at minimising loss in a switch. However, if the flow control loop does not go right up to the *real* source of the traffic (that is, the application), then avoiding switch cell congestion just results in the same traffic being queued at the UNI where the control loop is terminated. So, in the not unrealistic scenario of the ATM UNI being connected to another non-ATM network (LAN interconnection for example), we can expect significant queueing to occur in the ABR control unit.

We simulated for different buffer capacities to find an appropriate dimension for the rest of the studies. However, before presenting these results, we first look at what actually happens in the ingress buffer when traffic arrives, in order to have a conceptual model of the mechanics of the system.

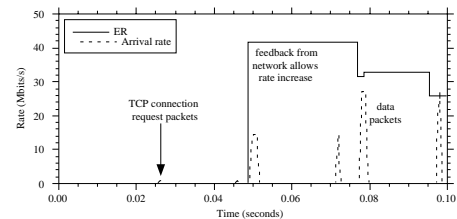
#### A. Transient behaviour

Figures 6(a) and 6(b) show, on two different timescales, what happens when a TCP connection is opened on LAN1, with a workstation on LAN2 as destination. Figure 6(a) also shows the number of active connections. The two curves plotted show the cell arrival rate at the ingress queue of LAN1's ABR control unit and the explicit rate granted to the same control unit by the network. We will call these two variables the arrival rate and the explicit rate in the following discussion. The arrival rate starts off at zero and the explicit rate starts off at the MCR (64 kbits/s in this example). At time 0.025 seconds, there are two barely visible blips on the arrival rate curve, which correspond to the arrival of TCP connection set-up handshaking packets. The arrival of these cells causes the ABR control unit to ask the network for the peak cell rate (145 Mbits/s). After a delay of around 0.03 seconds, the network responds to this request (i.e. a backwards RM cell arrives at the control unit), granting approximately 40 Mbits/s. The value of the bitrate is a result of the ERICA algorithm running in the switch along the route between LAN1 and LAN2. It was not possible to allocate the peak cell rate that was requested, since the traffic going through the switch shares the link capacity with the background traffic.

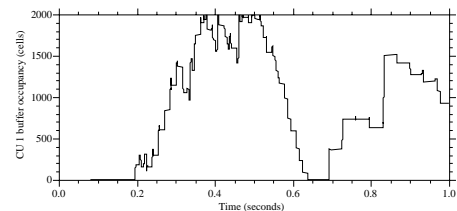
From this point on, we see that the explicit rate fluctuates around 40 Mbits/s, accommodating changes in the background arrival rate in the network. Figure 6(c) shows the buffer occupancy for control unit 1's ingress buffer. We see that after the connection has been established, the TCP traffic quickly increases, due to the expansion of the congestion window (slow-start mechanism) and the arrival of further connections. As a consequence, the buffer soon overflows (in this example, the buffer has a capacity of 2000 cells). However, once loss occurs, TCP at the sources which experienced loss halve their congestion windows and do congestion avoidance. This can be seen in the figure, where the



(a) Arrival rates and explicit rates



(b) First rate request



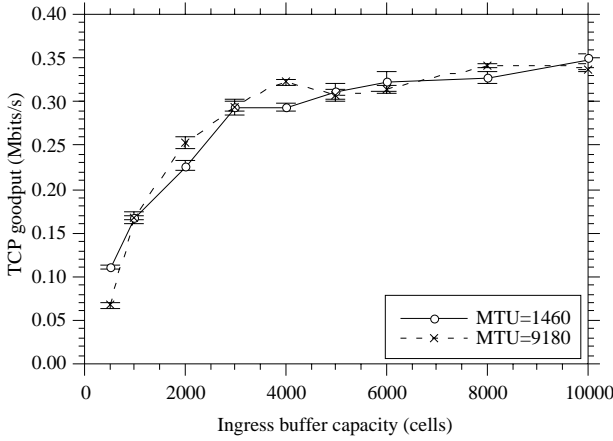
(c) Buffer occupancy trace

Fig. 6. Dynamics of the ingress control unit

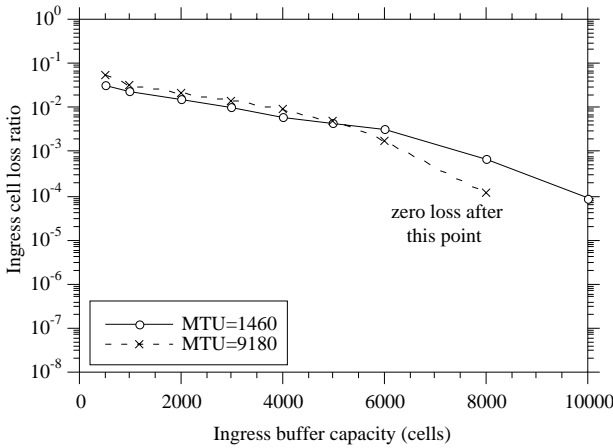
buffer occupancy starts to drop approximately 0.1 seconds after the first loss occurs.

#### B. Ingress buffer size

Figures 7(a) and 7(b) show the TCP goodput obtained per source and the cell loss ratio in the ingress buffer for different buffer sizes respectively. Curves are given for two different packet sizes: 1460 and 9180 bytes. We see that the best goodput is achieved for buffer sizes greater than 8000 cells. Increasing the buffer size above this point does not improve the performance since the loss is already zero or very low (the 'zero loss' legends in figure 7(b) mean that above these points, zero loss was observed, which is not plotable on a logarithmic scale). This plateau is around 70% percent of the theoretical maximum TCP goodput for both cases. For small buffer sizes (less than 2000 cells), smaller packets give slightly better performance as they result in smaller bursts arriving at the queue and lower the loss probability. For larger buffers, the increased efficiency of 9180 byte packets offers slightly better performance. In the rest of these studies, a buffer capacity of 8000 cells was adopted.



(a) TCP goodput



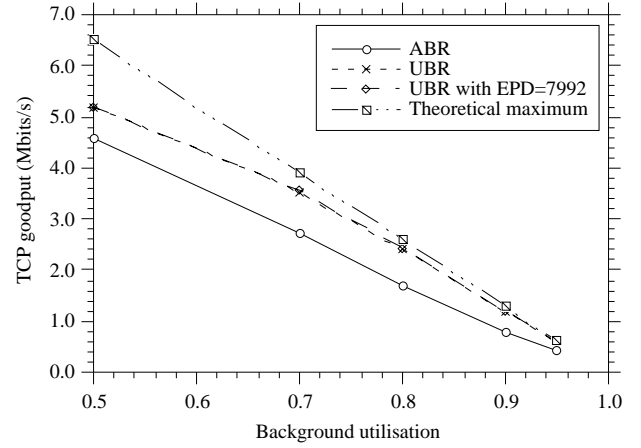
(b) Cell loss ratio

Fig. 7. Ingress buffer dimensioning

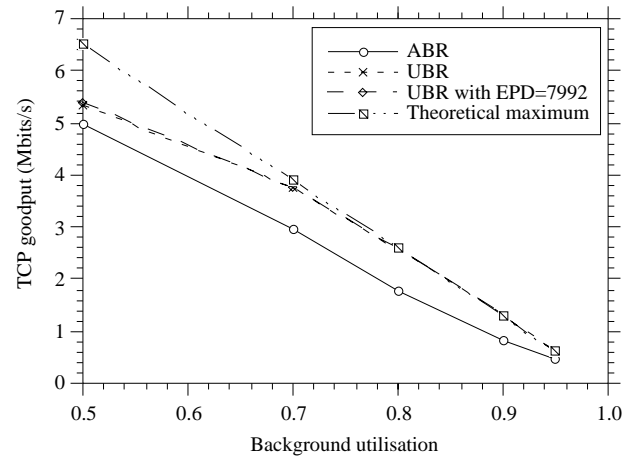
#### IV. VARYING BACKGROUND LOAD

The role of ABR in the network is to control the rate of ABR class traffic sources in the presence of non-ABR traffic so as to avoid cell loss in the network. To examine ABR's effectiveness at doing this, the TCP goodput over an ABR connection was measured when multiplexed with non-ABR background traffic. The background traffic used was described in section II-C. The TCP goodput obtained was compared with that obtained using UBR and UBR with EPD (for the EPD simulations, the threshold was set to be equal to the buffer size minus 200 cells, which was found to give the best performance for most cases).

We first consider a LAN case, where the total end-to-end propagation delay was 0.94 ms. Figures 8(a) and 8(b) show the TCP goodput per source obtained for a range of different background utilisations on the switch with ABR and UBR, with MTUs of 1460 and 9180 bytes. The control unit shaper size was 8000 cells (as recommended in the previous section). The curves also show the theoretical maxi-



(a) MTU of 1460 bytes

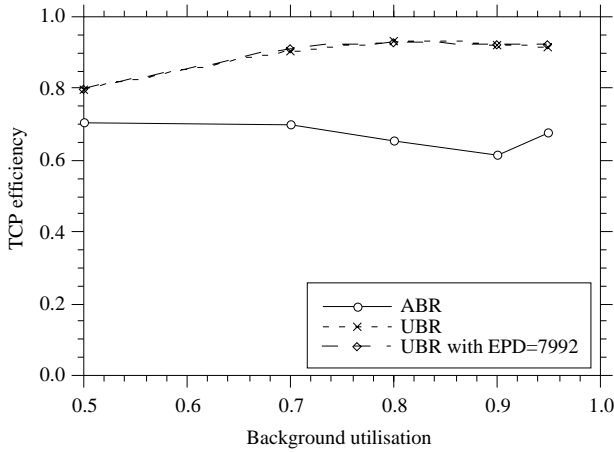


(b) MTU of 9180 bytes

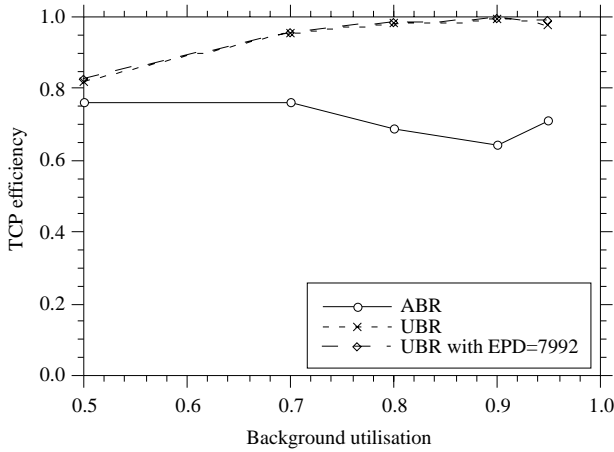
Fig. 8. TCP goodput per workstation for varying background load, with an ingress buffer size of 8000 cells in a LAN scenario

um TCP goodput, which is calculated by taking into account the overhead of the TCP/IP/AAL5/ATM protocol stack and the mean bandwidth available in the switch. Figures 9(a) and 9(b) show the TCP goodput efficiency in these same scenarios. The TCP efficiency is defined as the fraction of the theoretical maximum TCP goodput obtainable, taking into account all protocol overheads (except that of ABR RM cells). The theoretical maximum goodput used to obtain these graphs was based upon the link cell rate minus the observed background traffic rate.

We see that for the LAN case, using UBR for the interconnection VC gives better TCP goodput than with ABR, for MTUs of both 1460 bytes and 9180 bytes. The TCP efficiency increases with increasing background load for UBR, but not for ABR. Increasing the background load decreases the available bandwidth and thus decreases the bandwidth-delay product. This means that for low background loads, the bandwidth delay product is more likely to be greater

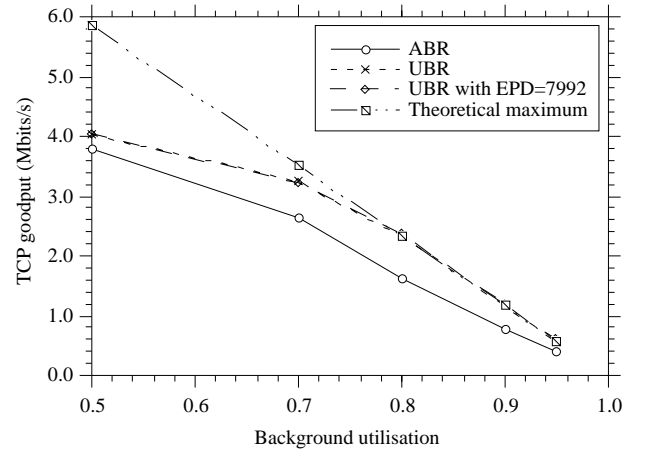


(a) MTU = 1460 bytes

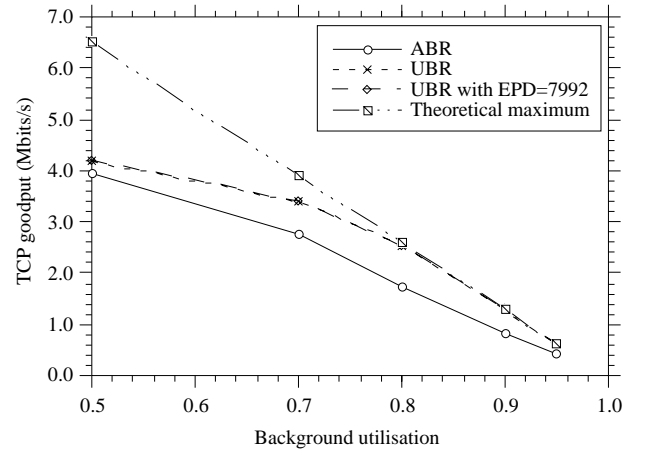


(b) MTU = 9180 bytes

Fig. 9. TCP efficiency for varying background load, with an ingress buffer size of 8000 cells and an MTU of 1460, in a LAN scenario



(a) MTU = 1460 bytes



(b) MTU = 9180 bytes

Fig. 10. TCP goodput per workstation for varying background load, with an ingress buffer size of 8000 cells in a MAN scenario

than the TCP congestion window, resulting in less efficient network utilisation. The efficiency curve for the ABR traffic is more or less flat, an upper bound being imposed by the ABR control loop. This means that the increase in efficiency with background load observed with UBR is not seen with ABR, since the efficiency is limited by the ABR control loop. Using EDP gives a slight performance increase over UBR, which is not visible on these curves.

We now consider the MAN case, where the end-to-end propagation delay was set to 10.9 ms. Results for TCP goodput and efficiency for MTUs of 1460 and 9180 are shown in figures 10(a), 10(b), 11(a) and 11(b). Figures 12(a), 12(b), 13(a) and 13(b) show the same results in the WAN scenario.

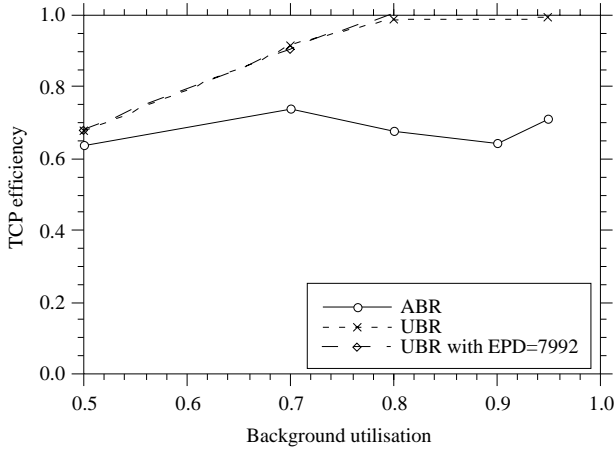
Again, we see that UBR gives better TCP goodput than ABR. When the background traffic is low, the performance degradation resulting from the increased bandwidth-delay product is evident. Finally we consider the WAN case, where the end-to-end propagation delay was set to 100.9 ms.

The results are shown in figures 12(a), 12(b), 13(a) and 13(b).

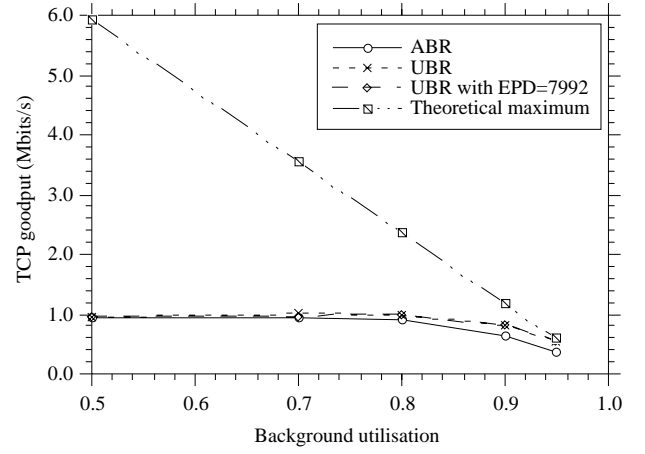
Tables II and III show the cell loss ratios in the ingress buffer (for ABR) and in the ABR/UBR buffer in the switch, with an MTU of 9180 bytes, for the LAN and MAN scenario respectively. No loss was observed in the WAN scenario, since the traffic is bottlenecked at the source by TCP since the bandwidth-delay product is greater than the socket buffer capacities. There is no loss in the ingress queue for UBR, since the queue's service rate is greater than the peak arrival rate.

We see that for UBR, increasing the background load increases the loss in the switch, whereas ABR successfully achieves its goal of avoiding loss in the ATM switch. Although there was always zero loss in the switch with ABR, loss occurred in the ingress buffer, at the edge of the ABR part of the network. Notice that for 95% load, the loss in the ingress buffer decreases slightly, even though the TCP

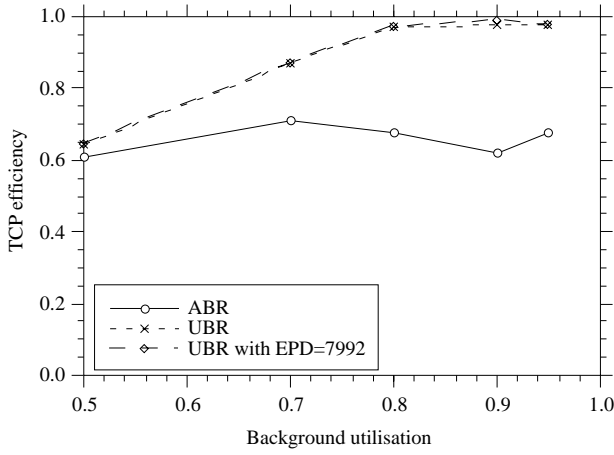




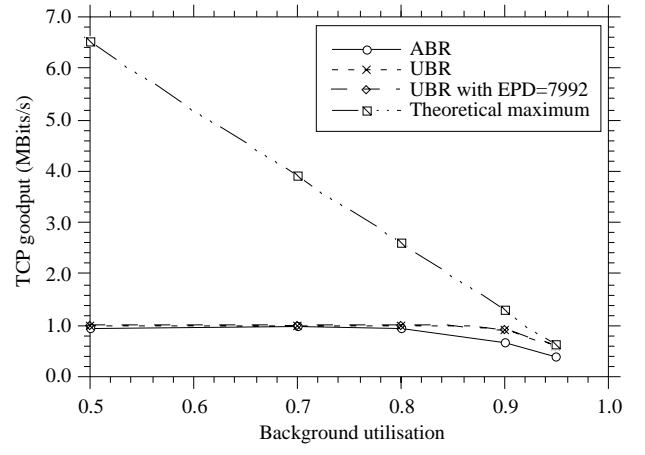
(a) MTU = 1460 bytes



(a) MTU = 1460 bytes



(b) MTU = 9180 bytes



(b) MTU = 9180 bytes

Fig. 11. TCP efficiency for varying background load, with an ingress buffer size of 8000 cells in a MAN scenario

Fig. 12. TCP goodput per workstation for varying background load, with an ingress buffer size of 8000 cells in a WAN scenario

$\mu_{bg}$	ABR		UBR
	ingress buffer loss ratio	switch loss ratio	switch loss ratio
0.50	$5.25 \times 10^{-6}$	0	$3.40 \times 10^{-6}$
0.70	$1.14 \times 10^{-5}$	0	$2.91 \times 10^{-5}$
0.80	$3.50 \times 10^{-4}$	0	$9.59 \times 10^{-5}$
0.90	$9.85 \times 10^{-4}$	0	$3.41 \times 10^{-4}$
0.95	$4.89 \times 10^{-4}$	0	$1.16 \times 10^{-3}$

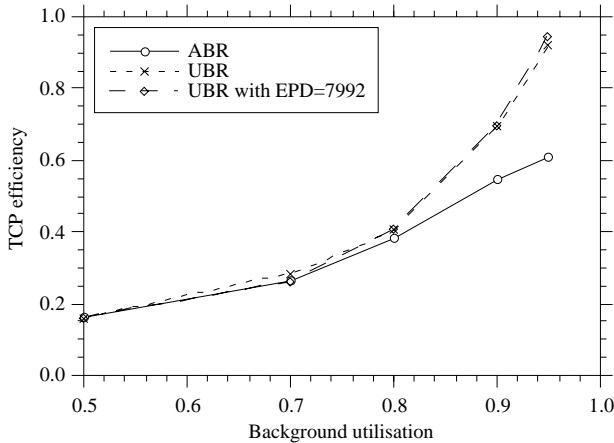
TABLE II

CELL LOSS RATIOS IN NETWORK FOR ABR AND UBR IN A LAN WITH AN MTU OF 9180 BYTES

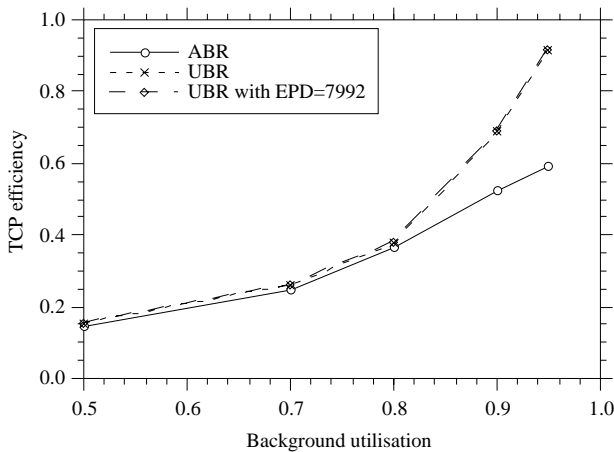
$\mu_{bg}$	ABR		UBR
	ingress buffer loss ratio	switch loss ratio	switch loss ratio
0.50	0	0	0
0.70	0	0	0
0.80	$3.06 \times 10^{-5}$	0	0
0.90	$3.38 \times 10^{-4}$	0	$4.91 \times 10^{-4}$
0.95	$4.24 \times 10^{-4}$	0	$5.74 \times 10^{-4}$

TABLE III

CELL LOSS RATIOS IN NETWORK FOR ABR AND UBR IN A MAN WITH AN MTU OF 9180 BYTES



(a) MTU = 1460 bytes



(b) MTU = 9180 bytes

Fig. 13. TCP efficiency for varying background load, with an ingress buffer size of 8000 cells in a WAN scenario

goodput continues to decrease. This illustrates the danger of adopting the cell loss ratio as a measure of the TCP goodput obtained.

This section has shown that TCP goodput was less when using an ABR VC for interconnection than with a UBR VC, for a simple one switch network. In the following section, we investigate why.

## V. ABR PERFORMANCE DEGRADATION

The fact that UBR performs well in these simulations is an important point here. ABR is an expensive feature, not only in terms of the hardware involved, but also because a system that requires the tuning of so many parameters for it to perform well requires a lot of engineering work.

It is therefore important to understand why better TCP performance was observed with UBR in the preceding study and to know in which scenarios and under what as-

sumptions this holds true. Compared to UBR, ABR can decrease TCP performance in four ways:

1. *RM cell overhead.* The bi-directional flow of RM cells reduces the bandwidth available for data traffic by 6.25%.
2. *Efficiency of the ABR control loop.* The source and destination algorithms and switch algorithms (in this case ERICA) can be badly tuned resulting in sub-optimal performance.
3. *Increased queueing delay.* In our model the traffic is queued at the edge of the ABR loop, resulting in one extra queueing stage which can only increase the overall delay. The ERICA algorithm aims to minimise this extra delay.
4. *Interaction between ABR and TCP.* ABR and TCP are two negative feedback control loops with very similar objectives. One could expect that embedding the ABR loop in the TCP loop could lead to undesirable interactions between the two [16].

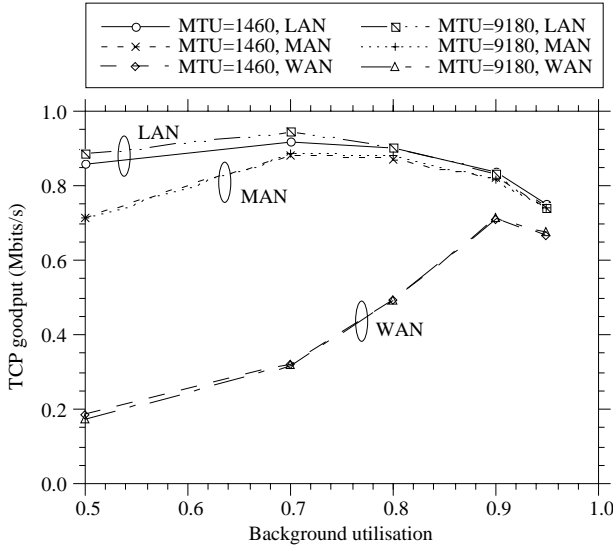
We will examine each of these points in turn and test the hypothesis that a particular phenomenon resulted in significant degradation of TCP goodput.

The reduction due to RM cells is fixed and is inevitable (there are some non-standard ABR-like mechanisms that do not use RM cells, such as the Gigaswitch from DEC, which are not considered here).

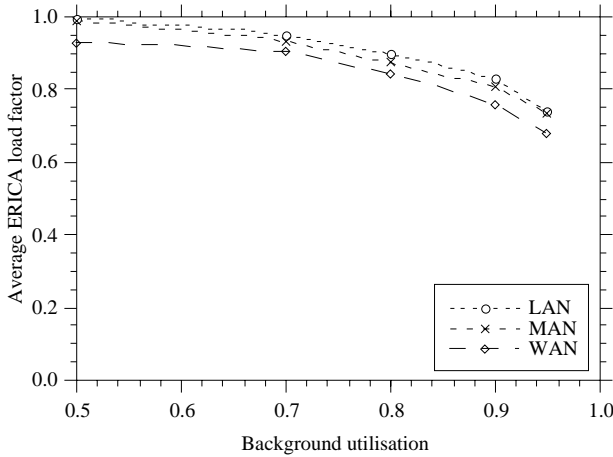
Next, let us consider the tuning of the ABR control loop. Explicit rate ABR, unlike binary mode, requires minimal parameter tuning at the sources<sup>3</sup>. Here we considered a very simple source strategy for requesting bandwidth (i.e. ask for the peak rate if there is something to send) and there are actually no parameters to tune. The ERICA algorithm running in the switch however, is much more complex with several parameters to be tuned. After experimentation, it was found that the parameters used (shown in table I) gave good performance. Figure 14(a) shows the ERICA load factors obtained in the switches for LAN, MAN and WAN cases with MTUs of 1460 and 9180 bytes. We can see that increasing the propagation delay decreases the load factor obtained, the traffic being limited not by ERICA but by TCP at the source. To verify this, figure 14(b) shows the ERICA load factors obtained using a greedy source instead of the LAN cluser. A greedy source always has something to send and allows us to isolate the performance of the ABR control loop.

We see that in general ERICA achieves very good load factors of almost 100%. For intense background traffic (greater than 80% of the link rate), the load factors decrease to between 70% and 80% of full utilisation. The longer the propagation delay, the greater the decrease in load factor for high background loads. Nevertheless, the algorithm performs very well under most background loads (the best performance resulted in 99.25% of the available bandwidth being used) and reasonably with extreme background loads.

<sup>3</sup>Note that for Network Interface Cards (NICs) to be able to run over binary-mode ABR switches, they must also implement the binary-mode rules. We do not consider binary-mode in this paper.



(a) TCP sources



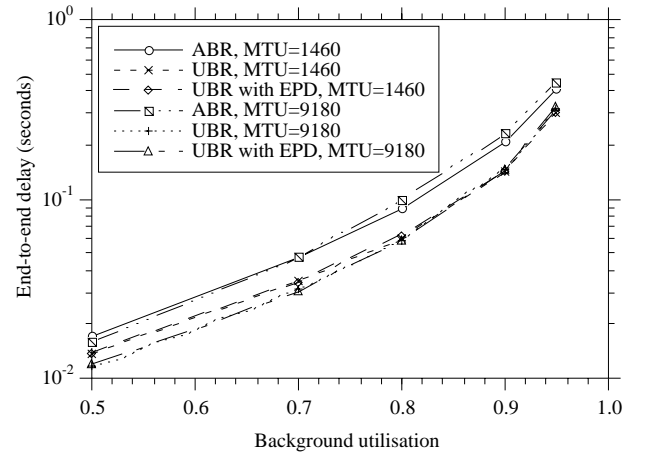
(b) greedy source

Fig. 14. ERICA load factor for LAN, MAN and WAN scenarios

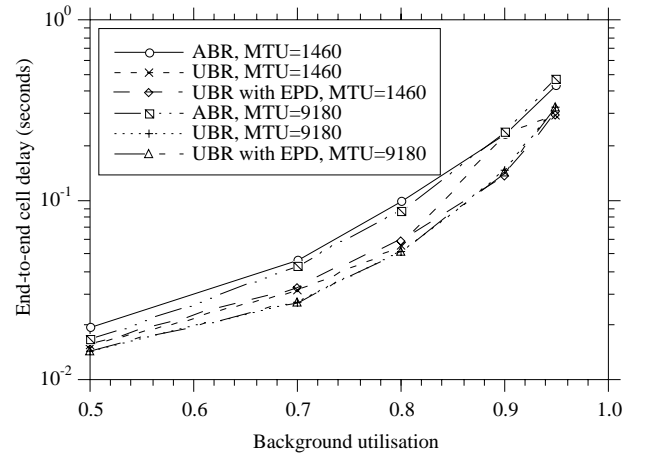
Now we examine the increase in queueing delay due to ABR. In our model, ABR adds an extra queueing stage between the source and destination, which in the best case introduces no extra delay, but in reality will always introduce some delay due to the sub-optimality of the ABR control loop. How this extra delay may influence performance is illustrated in figure 15.

Figures 16(a), 16(b) and 16(c) show the average cell delay measured from stations connected to LAN1 to their counterpart on LAN2.

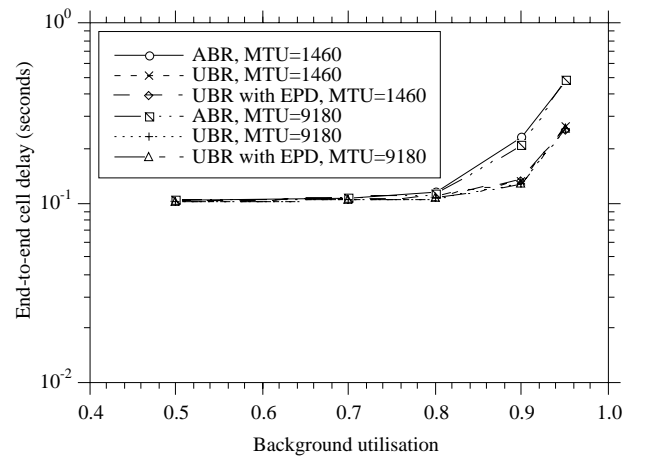
We see that ABR increases the network delay by a significant amount in all cases. The amount seems to be independent of the propagation delay in the network, but resulting from increased queueing delay. We see this in the WAN case, where there is negligible difference in the delay



(a) LAN



(b) MAN



(c) WAN

Fig. 16. End-to-end cell delay from LAN1 stations to LAN2 stations

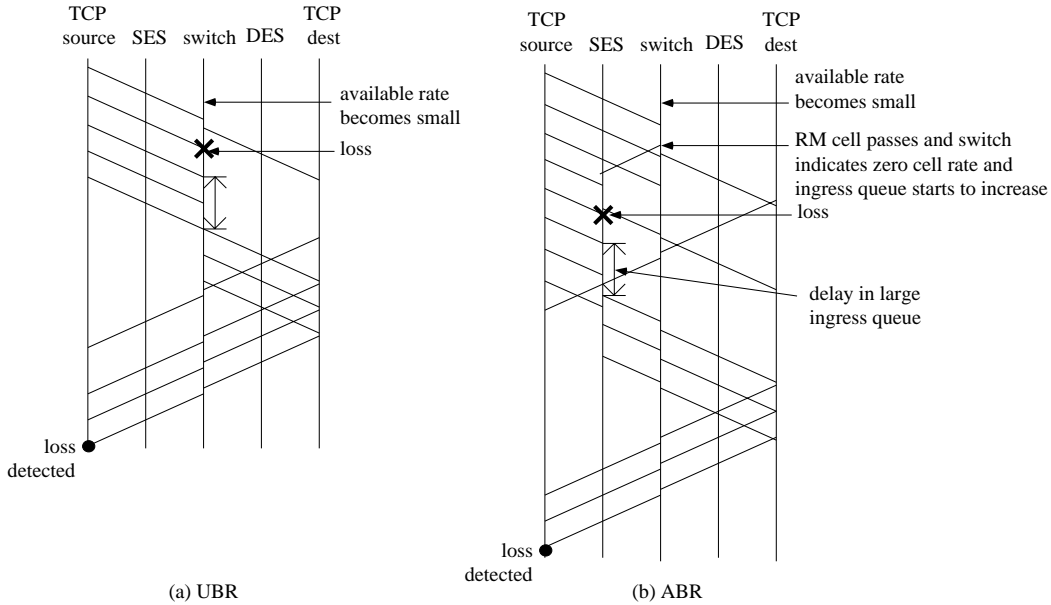


Fig. 15. How large ABR ingress buffers delay TCP feedback

when the background load is less than 80% (and the queueing delay consequentially less than the propagation delay of 100.9 ms). When the background load is greater than 80%, the delay is dominated by the queueing delay and we see a significant difference between the ABR and UBR curves. Here we can conclude that the introduction of the ABR control loop in our model increased the queueing delay in the network by a significant amount.

To conclude our investigation into the reasons for reduced TCP performance with ABR, we examine the interaction between the two congestion control loops. To do this we adjust the propagation delay in the ABR control loop ( $2 \cdot d_{net}$ ), while keeping the total end-to-end propagation delay,  $e$ , constant, i.e.  $2d_{net} + d_{switch} + d_{LAN} = c$  where  $c$  is a constant set to 50 ms. Since the delay experienced by the traffic consists of propagation delay and queueing delay, in the following we use the term ‘minimum delay’ to refer to the propagation delay part of an overall delay. Varying the minimum delay in the ABR loop while keeping the total minimum delay constant allows us to examine the influence of the ABR loop on the TCP loop. When  $d_{net} = 0$ , the minimum ABR feedback delay (i.e. the time between the switch modifying the ER field in a backwards RM cell and the rate change resulting from this modification being noticed at the switch) is zero and it therefore works at its most efficient, given the ERICA parameters it is working with it. It is at this point that TCP should dominate the dynamics of the traffic. When  $2 \cdot d_{net} = c$  then the minimum ABR feedback delay is the same as that for the TCP loop.

Figure 17 shows the influence of varying the delay in the ABR loop on the TCP goodput per workstation. We see that this has very little influence on the per-workstation TCP goodput, although there is a slight improvement in performance of around 0.1 Mbits/s when the ABR feedback delay is zero.

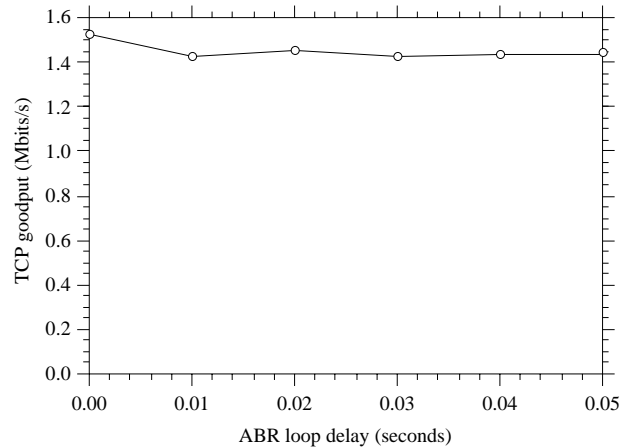


Fig. 17. Varying the ABR minimum feedback delay while keeping the TCP minimum feedback delay constant

The preceding results have not identified a single reason for the decrease in TCP goodput that resulted from introducing an ABR control loop. However, the RM cell overhead and the increased queueing delay play some role in decreasing TCP goodput, the combination of which results in a significant degradation. The interaction between the control loops has a very slight influence and, if badly tuned, the ABR control loop itself can decrease performance. In the remainder of this paper, we attempt to find some cases where ABR may improve TCP performance.

## VI. INFLUENCE OF OTHER NETWORK PARAMETERS ON PERFORMANCE

In this section, we attempt to increase our confidence in the hypothesis that TCP over UBR with adequate buffering offers better performance than TCP over ABR by ex-

amining its robustness with different network dimensions. More specifically, we change the number of TCP sources, the mean burst length of the background sources and the size of the ATM buffers. Increasing the number of TCP sources sharing a link decreases the average share of the bottleneck link per workstation. TCP should scale to this difference and we should still see good utilisation of the available bandwidth. Another consequence of increasing the number of sources is that the ‘density’ of traffic from one connection in a buffer decreases. In other words, the average time between two cells belonging to the same packet increases. This may have implications for EPD; we could expect that a larger number of connections may require a lower EPD threshold, since the number of simultaneously outstanding packets will be on average larger.

The only traffic in the network that is not influenced by the TCP protocol is the background traffic in the switches. This background traffic influences all of the network processes that influence TCP goodput (loss process, ACK arrival process and network delay process) and also influences the performance of the ERICA algorithm for ABR. We therefore try varying the background traffic characteristics to observe its influence on the relative performances of ABR and UBR.

Finally, we simulate the model with small buffers in the ATM switches, thus reducing the buffering in the UBR bottleneck node, but not in that of ABR. In such a scenario, ABR should give better performance than UBR, since it is able to exploit the buffering available in the end-systems.

#### A. Varying number of sources

First we consider the influence of the number of TCP sources on the mean TCP goodput. Recalling the model definition in section II, the number of sending stations on LAN 1 is  $N$ , the total number of workstations in the model being  $2 \cdot N$ . In this section we are referring to  $N$  when we say ‘number of workstations’. Figure 18 shows the TCP goodputs obtained for different numbers of sources, all sources using an MTU of 9180 bytes. The default MAN scenario was used ( $d_{net} = 5\text{ ms}$ ) with a background utilisation of 0.8. On the Y-axis, the graph shows the sum of TCP goodputs for all workstations on LAN 1, instead of the averaged per-workstation TCP goodput; this makes it easier to read as the number of sources increases.

Again, we see that UBR consistently out-performs ABR, achieving close to the theoretical maximum. EPD gave marginal performance increase, when the threshold was set to 7992 cells (which is barely visible in the graph).

#### B. Varying background traffic

Something that will have an important impact on the behaviour of the traffic in the ABR/UBR queue is the nature of the background traffic used. The background sources as a whole have four tuneable parameters: the number of sources, the mean rate, the peak rate and the mean burst duration. The influence of the mean rate has been addressed

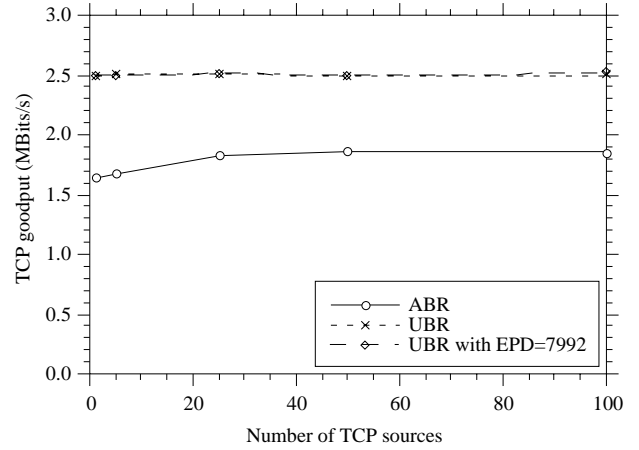


Fig. 18. TCP goodput per workstation for different numbers of sending workstations on a MAN

in previous sections and we do not investigate the influence of the number of sources. Here we tried different values for the mean burst length  $b_{bg}$ . Figure 19(a) shows the TCP goodput for a range of different mean burst lengths, with the peak rate set to 0.125 of the link rate (which makes the peak of the aggregate traffic 1.25 times the link rate) and the network delay  $d_{net}$  equal to  $10\text{ }\mu\text{s}$  (LAN scenario). Figure 19(b) shows the same results in the MAN scenario with  $d_{net} = 5\text{ ms}$ .

We see that the mean burst length of the on-off sources has very little influence on the observed TCP goodput and UBR always performs better than ABR.

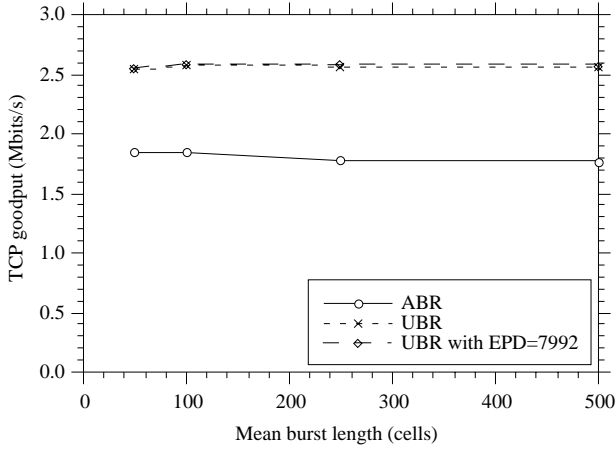
#### C. Switch buffer capacity

In this section we examine the performance of ABR and UBR when the switch has small buffers. Today’s ATM LAN switches have quite large buffers (around 8000 cells is typical). However, larger public network switches generally have much smaller buffers. Here we present results obtained when the buffers for ABR/UBR traffic in the switch had a capacity of 1000 cells. Figures 20(a), 20(b), 20(b) and 20(c) compares the TCP goodput for LAN and WAN scenarios. We see that when the ATM network is the bottleneck ( $\rho_{bg} > 0.65$ ), ABR always performed better than UBR. The difference was especially large in the LAN scenario.

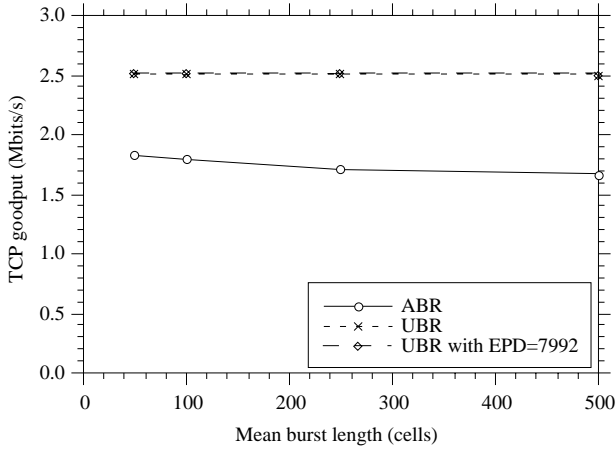
When we consider the *total* buffer capacity useable in the network, these results are not surprising, since ABR is able to buffer large bursts at the edge of the network, where it has an 8000 cell ingress buffer that UBR does not have. So, in the case where the traffic passes through at least one switch with small buffers, ABR may be useful in avoiding congestion in the network by buffering bursts at the ingress.

## VII. MULTI-HOP SCENARIOS

In the previous section we have considered a relatively simple network with only one switch. While this allowed us



(a) LAN

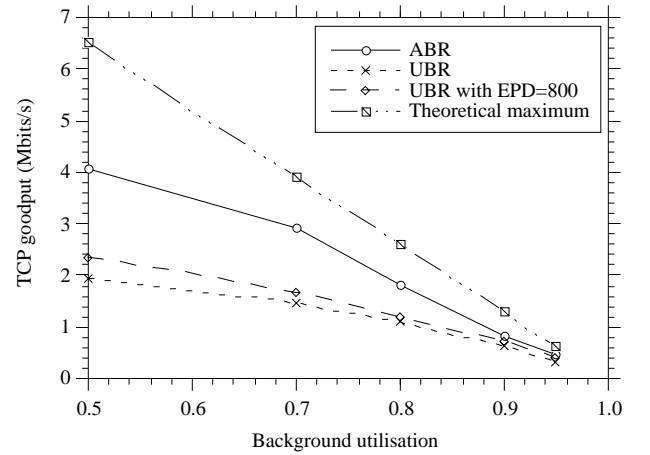


(b) MAN

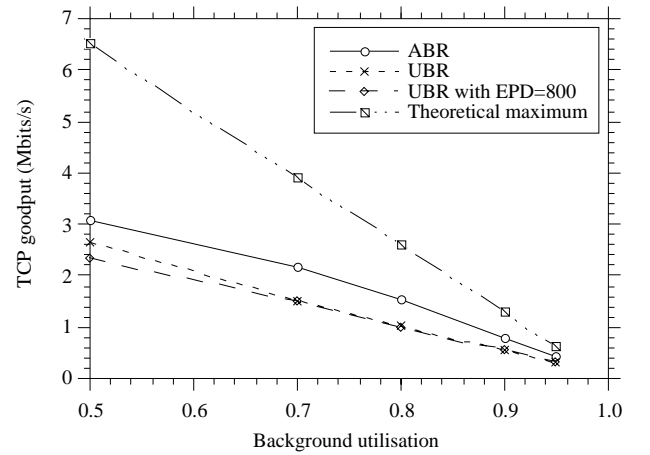
Fig. 19. TCP goodput per workstation for different mean background burst lengths and peak set to 0.125 times the link rate

to better understand the dynamics of the network, we also want to test our hypothesis in a more realistic, multi-hop network. The choice of the network studied in this section was based upon the desire to answer the following questions:

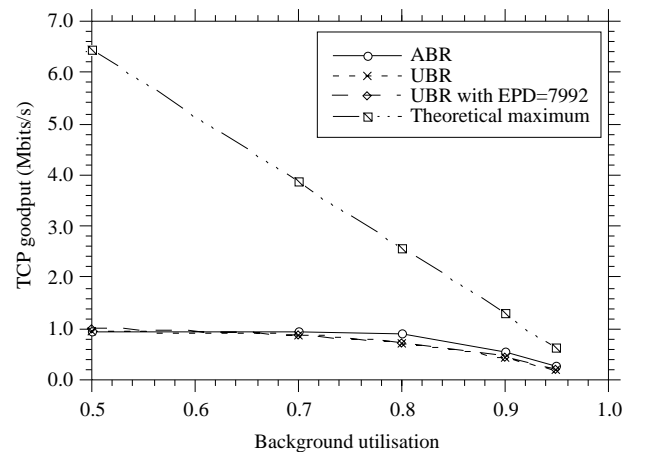
1. Does pushing congestion to the edge of the network improve performance for all TCP sources as a whole? If we consider a connection *A* multiplexed with a connection *B* at some node, connection *A* being bottlenecked in a downstream node that is not used by connection *B*, then ABR may improve performance for connection *B*, by reducing the traffic intensity of connection *A* rather than letting it be queued in the downstream node.
2. Does our original hypothesis hold true when VCs with different round-trip delays are multiplexed together?
3. Does ABR result in a different allocation of bandwidth in bottlenecked nodes?
4. Does EPD improve performance for nodes downstream of a congested switch?



(a) LAN



(b) MAN



(c) WAN

Fig. 20. TCP goodput per workstation for ABR and UBR in a WAN scenario with buffer capacities of 1000 cells in the switch output buffers

5. Does the presence of a topological loop result in a degradation of TCP goodput?
6. Does the presence of a topological loop result in performance degradation for VCs not passing through the bottleneck node.

To answer these question, we adopt two models: a ‘parking-lot’ configuration and a loop configuration.

#### A. Parking-lot configuration

The parking-lot model is shown in figure 21. There are 8 LANs, 6 of which have clusters of  $N$  source stations that start TCP connections with destination stations on another LAN. Traffic goes from one control unit to another over a VC which may be of type ABR or UBR (with or without EPD). Two traffic streams are multiplexed when they share the same output port of a switch. The same aggregations of 10 on-off sources used in section IV were used to create background traffic in switches 1 and 3. The mean background utilisations of switches 1 and 3 are denoted by  $\beta_1$  and  $\beta_2$  respectively.

The source and destination pairing is shown in table IV. The VC used to carry traffic from LAN  $x$  to LAN  $y$  is denoted by  $VC_{x,y}$ . LANs 1 and 6 symmetrically send data to one another over a wide-area connection (total end-to-end propagation delay of 53 ms), their bottleneck being either switch 1 or 3. Switch 1 has an 80% background utilisation. Two clusters of stations on LANs 2 and 3 both send data to stations on LAN6. This is a medium-distance connection, the propagation delay being 22.14 ms. These traffic flows are either bottlenecked in switch 3 or in switch 4, depending on  $\beta_2$ . We wish to see if our original hypothesis holds true when the VCs multiplexed together have different round-trip delays and whether ABR results in a different allocation of bandwidth between  $VC_{1,6}$ ,  $VC_{2,6}$  and  $VC_{3,6}$ .

Stations on LAN 4 send data to stations on LAN 5, which is a local area connection, the propagation delay being 80  $\mu s$ . For this traffic we would expect to see an improvement in performance when EPD is used, since this will reduce the amount of orphaned traffic in switch 4 resulting from congestion in switches 1 and 3. This traffic is always bottlenecked in switch 4.

Finally, traffic is created at LAN 7, destined for LAN 8. One might expect ABR to improve the performance for LAN 7 to LAN 8 traffic, since the traffic arriving at switch 1 from LAN 1 may be unnecessarily heavy bearing in mind that it will be bottlenecked downstream in switch 3. We wish to test whether ABR improves the performance of the LAN 7 to LAN 8 traffic by reducing the LAN 1 to LAN 6 traffic, or whether the TCP congestion control loop is sufficient for this purpose.

Results for the TCP goodput are shown in figures 22(a), 22(b), 22(c) and 22(d). The traffic from LAN 1 to LAN 6 is bottlenecked in switch 1 if  $\beta_2$  is less than 0.3, since the bandwidth available per-VC in switch 3 is  $145 \times 10^6(1 - \beta_2)/3$  and  $\beta_1$  is always 0.8.

From LAN	To LAN	VC	no. hops	bottle-neck	delay
LAN 1	LAN 6	$VC_{1,6}$	6	s1 or s3	53.02 ms
LAN 2	LAN 6	$VC_{2,6}$	6	s3 or s4	22.14 ms
LAN 3	LAN 6	$VC_{3,6}$	6	s3 or s4	22.14 ms
LAN 4	LAN 5	$VC_{4,5}$	4	s1	60.00 $\mu s$
LAN 6	LAN 1	$VC_{6,1}$	6	s1	53.02 ms
LAN 7	LAN 8	$VC_{7,8}$	4	s1	32.00 ms

TABLE IV  
TRAFFIC FLOW IN PARKING-LOT CONFIGURATION

Figure 22(a) shows the TCP goodput for LAN 1 to LAN 6 for ABR, UBR and UBR with EPD. We see that performance with UBR was significantly better than with ABR when switch 1 was the bottleneck, with EPD offering a slight additional increase. When switch 3 was the bottleneck (i.e.  $\beta_2 > 0.3$ ), ABR gave slightly better performance. Figure 22(b) shows the TCP goodput for LANs 2 and 3 to LAN 6. Here we observe that the TCP goodput was higher when ABR was used for the interconnection VCs. The mean goodputs for stations on LANs 1, 2 and 3 for ABR, UBR and UBR with EPD were 1.11 Mbits/s, 1.02 Mbits/s and 1.02 Mbits/s respectively. Although there was a slight improvement in performance for ABR, the overall goodput remained much the same, indicating a shift in allocation of bandwidth to different VCs at switch 3. ABR favoured the traffic from LANs 2 and 3, since it tried to equally share the available bandwidth between the three VCs multiplexed in switch 3. With ABR, traffic from LAN 1 was sometimes told to slow down, in favour of the LAN 2 and LAN 3 traffic, when it would not otherwise do so. This resulted in a reduction of LAN 1 traffic and consequently an increase in LAN 2 and LAN 3 traffic. This demonstrates ABR’s ability to improve fairness in the network.

Figure 22(c) shows the TCP goodput for the LAN 4 to LAN 5 traffic. As  $\beta_2$  increases, the goodput for  $VC_{4,5}$  increases, since the traffic on  $VC_{1,6}$ ,  $VC_{6,1}$ ,  $VC_{2,6}$  and  $VC_{6,2}$  is bottlenecked in switch 3. The noticeable feature of this graph is the increase in performance when using UBR with EPD. This is due to the reduction of ‘orphaned’ cells passing through switch 4 resulting from loss in upstream nodes (switches 1 and 3) and illustrates EPD’s usefulness in reducing resource waste.

Figure 22(d) shows the goodput for the LAN 7 to LAN 8 traffic. One may expect ABR to improve performance for this traffic stream when  $\beta_2 > 0.3$  by slowing down  $VC_{1,6}$ . Traffic on this VC is bottlenecked in switch 3. With UBR, significant loss may consequently occur in switch 3 which will result in a certain amount of unnecessary traffic in switch 1 (unnecessary because it will be lost downstream). ABR, however, will slow down  $VC_{1,6}$  to avoid congestion in switch 3, this leaving more resources available for  $VC_{7,8}$ . TCP will do the same thing on an end-to-end instead of a node-to-node basis, but more slowly. Figure 22(d) shows, however, that there is no improvement in performance with

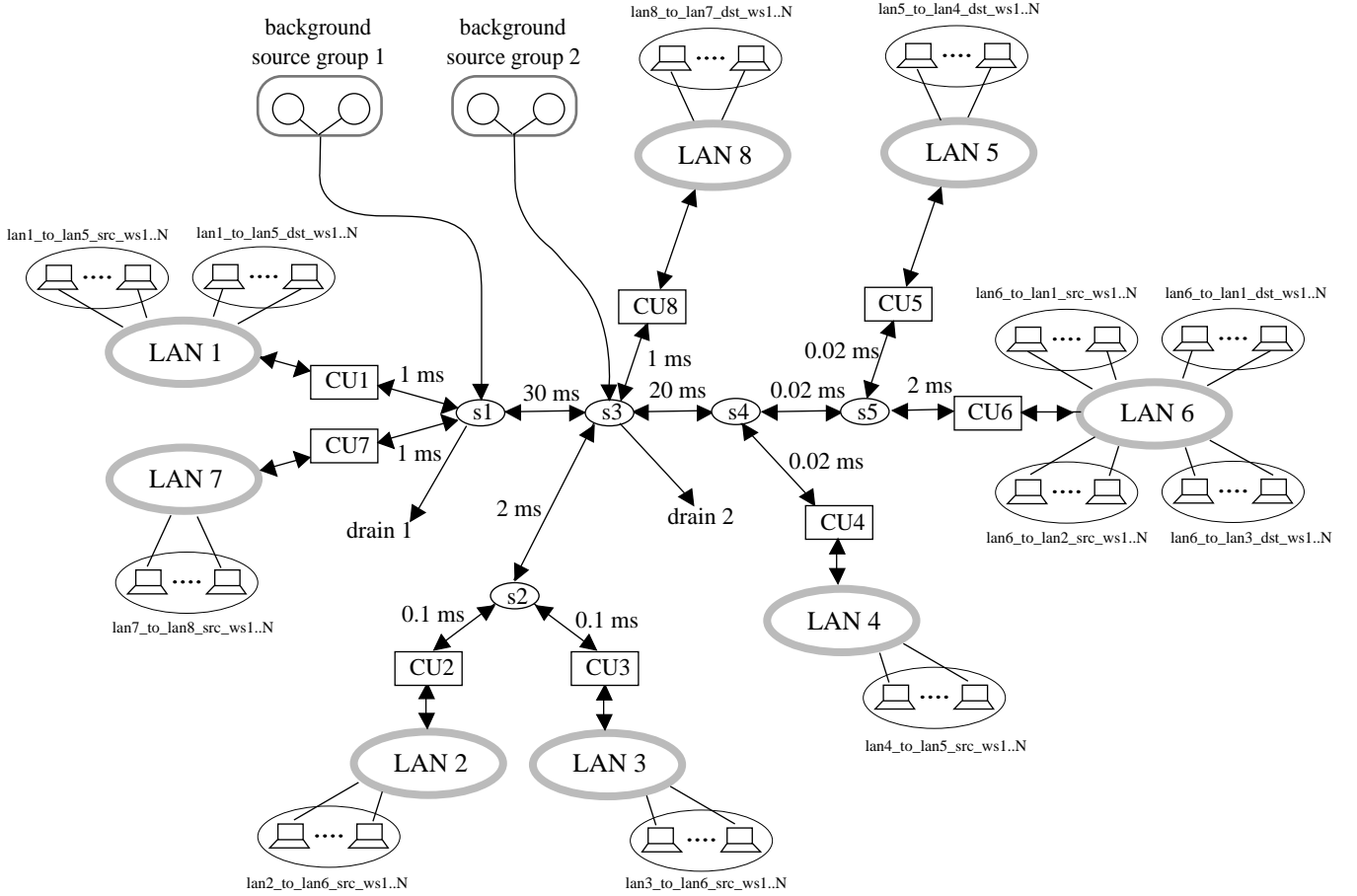


Fig. 21. WAN scenario

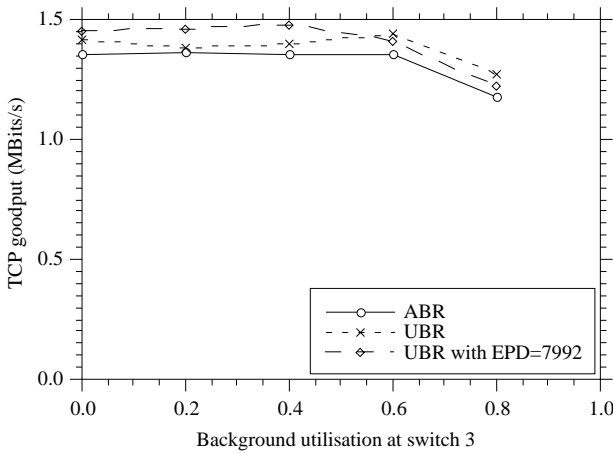


Fig. 23. TCP goodput for all sources in the parking-lot scenario

ABR. Although ABR may indeed be avoiding congestion in switch 1 better than TCP does, this does not compensate for the additional overheads of ABR discussed in section III.

Finally, figure 23 shows the average goodput for all workstation in the model. We see that UBR and UBR with EPD give better TCP goodput than ABR, in a scenario where several VCs with different round-trip delays are multiplexed together over several nodes. In answer to the first

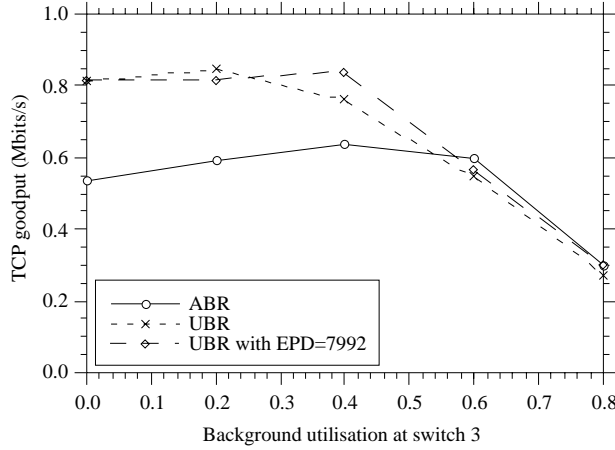
four questions posed at the start of this section, we can say that:

- pushing congestion back to the edges of the network did not improve overall TCP performance,
- our original hypothesis holds true when several VCs with different round-trip times are multiplexed together.
- the bandwidth allocation between VCs can be controlled using ABR and
- EPD improves performance in switches downstream of a congested switch

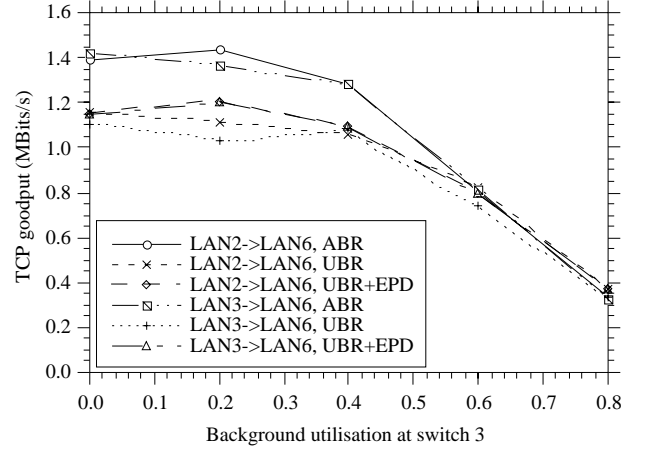
### B. Loop configuration

In this section we consider the performance over ABR and UBR VCs in a network with a loop configuration, shown in figure 24. There are three VCs providing interconnection between LANs 1 and 2, LANs 2 and 3 and LANs 3 and 1. The traffic flow is summarised in table V. The important feature of this network is that the interaction between two VCs that are multiplexed together propagates around the network loop. For example, LAN 1 to 3 traffic is multiplexed with LAN 2 to LAN 1 traffic in the output port of switch 2 that leads to switch 3. The LAN 2 to LAN 1 traffic is then multiplexed with, and consequently influences, the LAN 3 to LAN 2 traffic in switch 3. This traffic flow is then multiplexed with the LAN 1 to LAN 3 traffic in

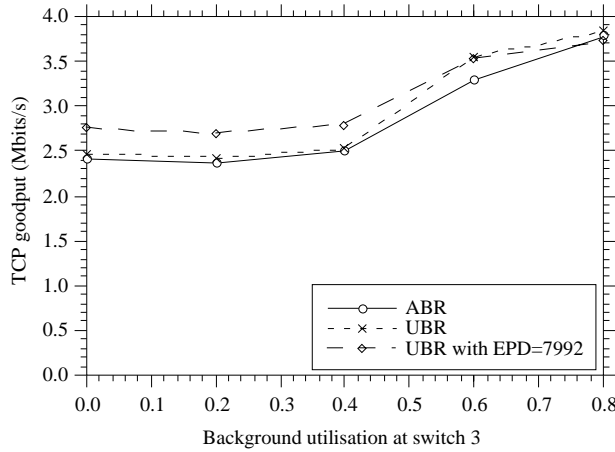




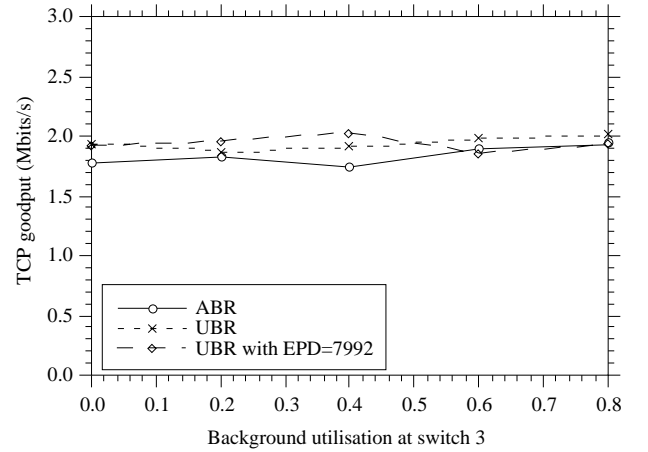
(a) LAN 1 to LAN 6



(b) LAN 2 and LAN 3 to LAN 6



(c) LAN 4 to LAN 5



(d) LAN 7 to LAN 8

Fig. 22. TCP goodput per workstation in the parking-lot scenario

From LAN	To LAN	path	VC
1	3	$CU1 \rightarrow s1 \rightarrow s2 \rightarrow s3 \rightarrow CU3$	$VC_{1,3}$
2	1	$CU2 \rightarrow s2 \rightarrow s3 \rightarrow s1 \rightarrow CU1$	$VC_{2,1}$
3	2	$CU3 \rightarrow s3 \rightarrow s1 \rightarrow s2 \rightarrow CU2$	$VC_{3,2}$

TABLE V  
TRAFFIC FLOW IN LOOP CONFIGURATION

switch 1 and the loop is completed. We call this loop the ‘multiplex loop’. In such a configuration, it is possible that the ABR and TCP control loops controlling the traffic on the VCs combined with the multiplex loop could lead to performance degradation. We wish to test this hypothesis and test whether the performance for  $VC_{2,1}$ , which does not pass through switch 1 is influenced, via the multiplex loop, by the background load on switch 1.

Figures 25(a), 25(b) and 25(c) show the TCP goodput for

stations on LANs 1, 2 and 3 respectively. We see that the loop configuration does not introduce any influence on the results for TCP goodput. As expected, the goodput for LANs 1 and 3 decreases as the background load increases. For  $\mu_{bg} < 0.6$  there is no influence on the goodput, since TCP itself is the bottleneck, due to the window size and round-trip time. When  $\beta_2 = 0.5$ , the TCP goodputs per station for UBR for LANs 1 and 3 are 1.8 Mbits/s and 1.75 Mbits/s respectively. This gives a total goodput for all 20 workstations whose traffic goes through switch 1 of 35.5 Mbits/s. This is close to the goodput obtained when the sources on LAN 2 are removed (thus removing the multiplexing loop). We also observe that the multiplex loop has no influence on the TCP goodput for sources on LAN 2.

## VIII. CONCLUSION

We tested the hypothesis that provided that there is adequate buffer capacity in the ATM network, better overall

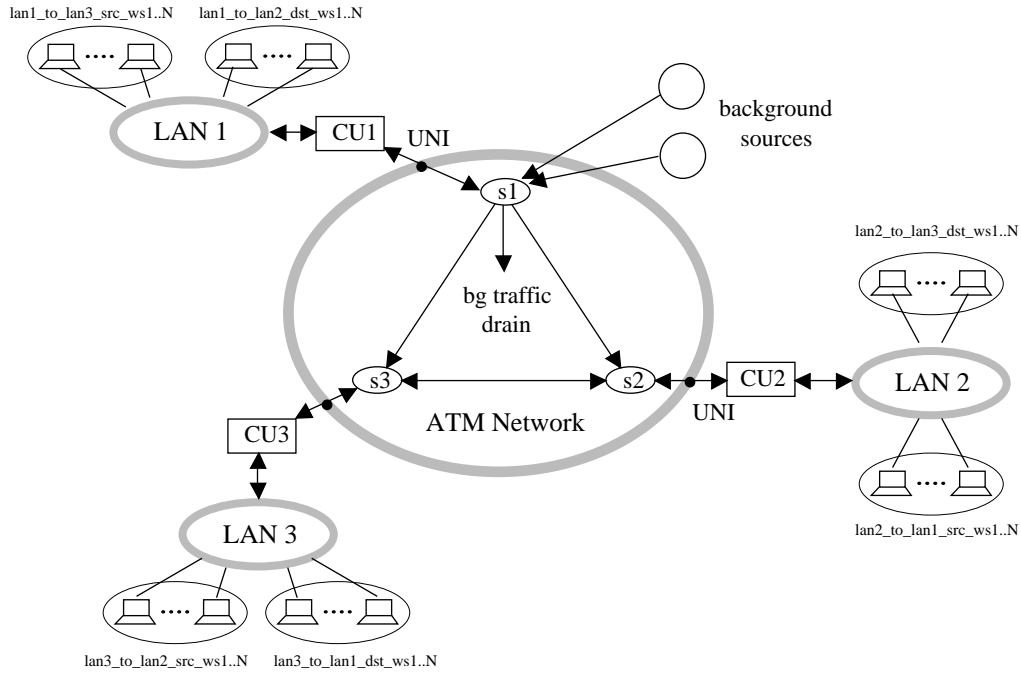


Fig. 24. Loop network

TCP performance is obtained when using a UBR VC to interconnect LANs than when using ABR. We found this to be true in a wide range of cases, with the performance degradation observed with ABR being around 10%. We identified four phenomena that can lead to degradation of ABR performance and tested each of these for a significant influence. We found the reduction in TCP goodput to be due to the extra queueing delay incurred in the ABR end-system and the overhead of RM cells. The interaction between the nested TCP and ABR congestion control loops was found to cause only a very small degradation of performance. Using EPD for the UBR service was shown to improve performance in some cases, but raw UBR still worked well. We further tested whether pushing the congestion to the edges of the network using ABR improved performance in a parking-lot scenario and whether the hypothesis held true in the presence of a multiplex loop. In both cases we were unable to disprove our hypothesis. We showed, however, how ABR allows the network to control fairness between VCs, a feature that may be desirable for LAN interconnection services.

When the buffer capacity in the ATM network is small compared to the buffering available in the end-systems, then ABR can significantly improve TCP goodput by avoiding loss in the ATM switches. In these studies, ABR performed up to 100% better than UBR when the switches had buffers of 1000 cells and the ABR end-systems 8000 cells.

We conclude that from the point of view of overall TCP goodput for LAN interconnection, UBR offers better performance than ABR providing that there is adequate buffering in the ATM network, 8000 cells being sufficient for all cases studied here. An important feature of ABR, how-

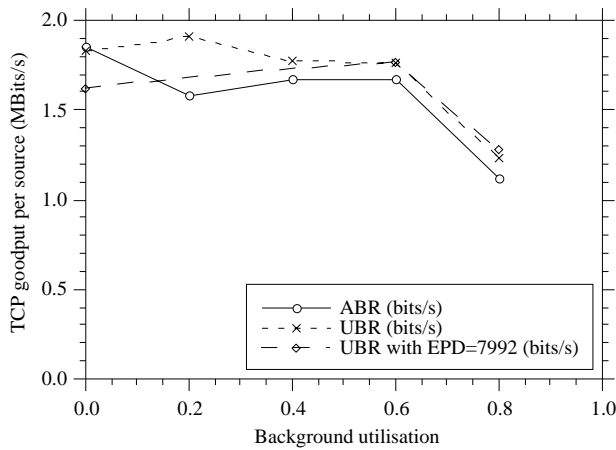
ever, is that it allows the interconnection service provider to control fairness between traffic streams, which would not otherwise be possible.

## IX. ACKNOWLEDGEMENT

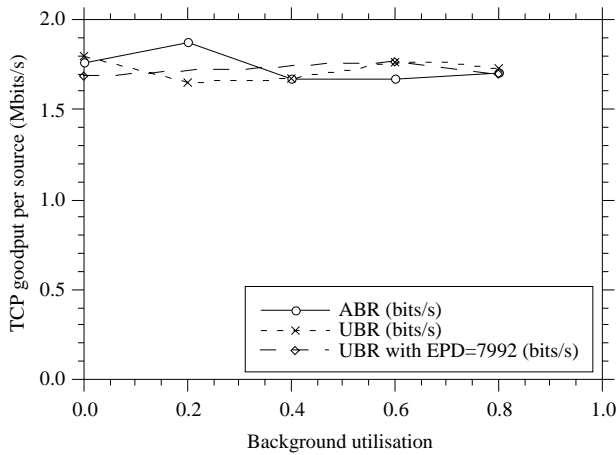
The authors acknowledge the support of the Swiss PTT Telecom for funding this research. They also thank Olivier Bonaventure and Andres Meyer for constructive feedback on the STCP tool.

## REFERENCES

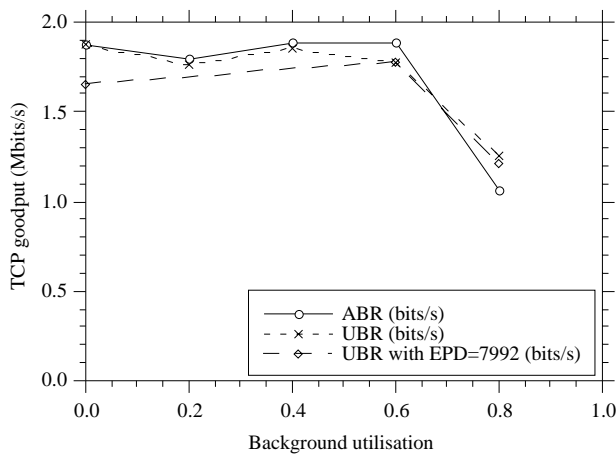
- [1] R. Stevens, *TCP/IP Illustrated: The Protocols*, vol. 1. ISBN-0-201-63346-9: Addison-Wesley, 1994.
- [2] S. Manthorpe, *Implications of the Transport Layer for Network Dimensioning*. PhD thesis, EPFL, 1997. (to appear).
- [3] M. Garrett and W. Willinger, "Analysis, modeling and generation of self-similar VBR video traffic," in *Proceedings of ACM SigComm, London*, Sept. 1994.
- [4] R. Grünfelder, J. P. Cosmas, S. Manthorpe, and A. Odinma-Okafor, "Characterization of video codecs as autoregressive moving average processes and related queueing system performance," *IEEE Journal on Selected Areas in Communications*, vol. 9, pp. 284-293, Apr. 1991.
- [5] ATM Forum, *ATM Forum Traffic Management Specification version 4.0, revision 10*, February 1996.
- [6] J. Roberts, ed., *Broadband Networks Teletraffic (Final report of Action COST242)*. Springer, 1996.
- [7] R. Jain, S. Kalyanaraman, and R. Viswanathan, "The OSU scheme for congestion avoidance using Explicit Rate Indication," ATM Forum document af94-0883, Ohio State University, Sept. 1994.
- [8] R. Jain, S. Kalyanaraman, R. Goyal, S. Fahmy, and R. Viswanathan, "AF-TM-1172: The ERICA Switch Algorithm: A Complete Description," tech. rep., ATM Forum, August 1996.
- [9] Y. Zhao, S. qi Li, and S. Sigarto, "A Linear Dynamic Model for Design of Stable Explicit-Rate ABR Control Schemes," ATM Forum Document af96-0606, University of Texas at Austin, Apr. 1996.



(a) LAN1



(b) LAN2



(c) LAN3

Fig. 25. TCP goodput per workstation in loop scenario

- [10] A. Romanow and S. Floyd, "Dynamics of TCP traffic over ATM networks," *IEEE Journal of Selected Areas in Communication*, 1994.
- [11] S. Manthorpe, "Buffering and Packet Loss in the DQDB to ATM Interworking Unit," in *Proceedings of the Integrated Broadband Communications Networks and Services Conference, Copenhagen*, (ISBN 0-444-81584-8), April 1993.
- [12] R. Vogt, U. Killat, J. Ottensmeyer, and M. Rümekasten, "A Concept for Interconnecting DQDB MANs through ATM-based B-ISDN and Related Issues with respect to Simulation," in *Proceedings of Interworking 92, Bern*, November 1992.
- [13] S. Manthorpe, "STCP User Manual," Tech. Rep. 1.0, LRC-DI-EPFL, <http://lrcwww.epfl.ch/~manthorp/stcp>, 1996.
- [14] G. Wright and R. Stevens, *TCP/IP Illustrated: The implementation*, vol. 2. 0-201-63354-X: Addison-Wesley, 1995.
- [15] V. Jacobsen and M. Karels, "Congestion Avoidance and Control," in *Proceedings of SIGCOM '88, Stanford*, August 1988.
- [16] S. Jagannath and N. Yin, "End-to-end TCP performance in IP/ATM networks, af96-1711," ATM Forum contribution AF96-1711, ATM Forum, Dec. 1996.